

```

RRRRRRRRRRRRRRR      MMM      MMM      SSSSSSSSSSSSS
RRRRRRRRRRRRRRR      MMM      MMM      SSSSSSSSSSSSS
RRRRRRRRRRRRRRR      MMM      MMM      SSSSSSSSSSSSS
RRR      RRR      MMMMMM      MMMMMM      SSS
RRR      RRR      MMMMMM      MMMMMM      SSS
RRR      RRR      MMMMMM      MMMMMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRRRRRRRRRRRRRR      MMM      MMM      SSSSSSSSSSS
RRRRRRRRRRRRRRR      MMM      MMM      SSSSSSSSSSS
RRRRRRRRRRRRRRR      MMM      MMM      SSSSSSSSSSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      MMM      SSS
RRR      RRR      MMM      MMM      SSSSSSSSSSSSS
RRR      RRR      MMM      MMM      SSSSSSSSSSSSS
RRR      RRR      MMM      MMM      SSSSSSSSSSSSS

```

52

Syn

NTS
NTS

NTS

NTS
NTS

NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTS

NTS

NTS
NTSNTS
NTSNTS
NTS

NTS

NTS
NTSNTS
NTSNTS
NTSNTS
NTS

NTS

NT
NT

NTS

NT
NT

NT

PI

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```


(2) 57
(3) 84

DECLARATIONS
RMSS\$MODIFY - \$MODIFY ROUTINE

```
0000 1      $BEGIN RMSOMODFY,000,RMSRMS,<MODIFY FUNCTION>
0000 2
0000 3
0000 4      *****
0000 5      *
0000 6      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8      *  ALL RIGHTS RESERVED.
0000 9      *
0000 10     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15     *  TRANSFERRED.
0000 16     *
0000 17     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19     *  CORPORATION.
0000 20     *
0000 21     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23     *
0000 24     *
0000 25     *****
0000 26
0000 27     ++
0000 28     Facility: RMS32
0000 29
0000 30     Abstract:
0000 31             This module performs the $MODIFY function.
0000 32
0000 33     Environment:
0000 34             Star processor running Starlet exec.
0000 35
0000 36     Author: L. F. Laverdure      Creation Date: 21-JUN-1977
0000 37
0000 38     Modified By:
0000 39
0000 40             V03-002 RAS0120      Ron Schaefer      25-Jan-1983
0000 41             Add echo SYSS$INPUT to SYSS$OUTPUT modify function.
0000 42
0000 43             V03-001 KBT0186      Keith B. Thompson  23-Aug-1982
0000 44             Reorganize psects and rename entry point to single '$'
0000 45
0000 46             V02-005 RAS0018      Ron Schaefer      9-Aug-1981
0000 47             Fix broken ASSUME caused by stream files.
0000 48
0000 49             V02-004 MCN0001      Maria del C. Nasr  29-Jul-1981
0000 50             Rename entry point to RMS$$ to support long branches.
0000 51
0000 52             V02-003 REFORMAT      K. E. Kinnear     31-Jul '80      9:46
0000 53
0000 54     --
0000 55
```



```
0000 57      .SBTTL  DECLARATIONS
0000 58
0000 59      :
0000 60      : Include Files:
0000 61      :
0000 62      :
0000 63      :
0000 64      : Macros:
0000 65      :
0000 66
0000 67      $IFBDEF
0000 68      $FABDEF
0000 69      $PSLDEF
0000 70      $RMEDEF
0000 71      $RMSDEF
0000 72
0000 73      :
0000 74      : Equated Symbols:
0000 75      :
0000 76
00000020 0000 77      FOP=FAB$L_FOP*8      ; bit offset to fop
0000 78
0000 79      :
0000 80      : Own Storage:
0000 81      :
0000 82
```



```
0000 84      .SBTTL  RMS$MODIFY - $MODIFY ROUTINE
0000 85
0000 86      :++
0000 87      : RMS$MODIFY -- Modify Routine.
0000 88
0000 89      : This routine performs the $modify processing.
0000 90      : It has one function:
0000 91      :   To provide an 'escape' mechanism to perform non-standard
0000 92      :   rms functions.
0000 93
0000 94      : The functions currently implemented are:
0000 95      :   1. To rewrite modified file attributes.
0000 96      :   2. To enable/disable echoing of SYS$INPUT to SYS$OUTPUT.
0000 97
0000 98      : Calling Sequence:
0000 99
0000 100     : Entered from exec as a result of user's calling SYS$MODIFY
0000 101     : (e.g., by using the $modify macro).
0000 102
0000 103     : Input Parameters:
0000 104
0000 105     :   AP      user's argument list addr
0000 106
0000 107     : Implicit Inputs:
0000 108
0000 109     : The contents of the fab and possible related user interface
0000 110     : blocks.
0000 111     : The esc bit is set in fop indicating that the caller desires
0000 112     : to execute one of the 'escape sequences', otherwise known as
0000 113     : 'back doors' or 'kludges', that is, ways of tricking rms into
0000 114     : thinking that the situation is other than rms's current view of it.
0000 115     : These will, hopefully, remain few in number. Implementing these
0000 116     : as a service is necessary due to the requirement for exec mode
0000 117     : privileges and additionally gives us a handle on the extent of the
0000 118     : cancer. Improper use of an escape sequence can blow rms out of the
0000 119     : water.
0000 120
0000 121     : Output Parameters:
0000 122
0000 123     :   R0      status code
0000 124     :   R1      destroyed
0000 125
0000 126     : Implicit Outputs:
0000 127
0000 128     : The ifab and all related internal rms structures are modified
0000 129     : as per the requirements of the operation.
0000 130     : FAB$L_STS and FAB$L_STV
0000 131
0000 132     : A completion ast is queued if so specified by the user.
0000 133
0000 134     : Completion Codes:
0000 135
0000 136     :   Standard rms (see functional spec for list).
0000 137
0000 138     : Side Effects:
0000 139
0000 140     :   Dependent upon the type of modify.
```



```
0000 141 :  
0000 142 :--  
0000 143 :  
FFFD' 30 0000 144 $ENTRY RMS$MODIFY  
0000 145 BSBW RMS$FSET ; do common setup  
0003 146 ; note: does not return on error  
0003 147 RMSERR ENV ; assume failure  
0008 148  
0008 149 :  
0008 150 : Check for modify of 'escape' type and branch if bit not set.  
0008 151 :  
0008 152 :  
09 68 3B E1 0008 153 BBC #FAB$V_ESC+FOP,(R8),MODXIT ; branch if not 'escape'  
000C 154 CASE TYPE=W,- ; low word of context field  
000C 155 LIMIT=#RMESC_SETRFM,-  
000C 156 SRC=FAB$L_CTX(R8),-  
000C 157 DISPLIST=-  
000C 158 <SETRFM,- ; RMESC_SETRFM  
000C 159 PPFECHO> ; RMESC_PPFECHO  
FFEB' 31 0015 160  
0015 161 MODXIT: BRW RM$EXRMS  
0018 162
```

```
0018 164
0018 165 :++
0018 166 : Escape type one - set rfm
0018 167 :
0018 168 : Inputs:
0018 169 :
0018 170 :     rfm,mrs, and fsz (if vfc)
0018 171 :
0018 172 : Outputs:
0018 173 :
0018 174 :     Related ifab fields are changed to values specified by inputs.
0018 175 :
0018 176 : Notes:
0018 177 :
0018 178 :     1. User is responsible for saving the previous contents of the
0018 179 :        rfm, mrs, and fsz fields if needed for later restore.
0018 180 :
0018 181 :     2. If the file is accessed for put, final attributes written
0018 182 :        to the file on close will be those currently in effect.
0018 183 :
0018 184 :     3. There are no default values for any of the input fields.
0018 185 :
0018 186 :     4. If setting rfm to udf and not block i/o accessed, results
0018 187 :        are unpredictable.
0018 188 :
0018 189 :     5. If setting rfm to fix and mrs is 0, an error is generated
0018 190 :        but further rms calls will produce unpredictable results.
0018 191 : --
0018 192 :
0018 193 SETRFM:
0018 194
0018 195 RMSERR RFM ; anticipate problems
0018 196 CMPB FAB$B_RFM(R8),#FAB$C_MAXRFM; within range?
0018 197 BGTRU MODXIT ; branch if not
0018 198
0018 199 CMPB FAB$B_RFM(R8),#FAB$C_VFC
0018 200 BNEQ 10$ ; branch if not vfc format
0018 201 MOVW FAB$B_FSZ(R8),IFB$B_FSZ(R9); set fsz
0018 202 MOVW FAB$B_RFM(R8),IFB$B_RFMORG(R9); set rfm
0018 203 MOVW FAB$B_MRS(R8),IFB$B_MRS(R9); set mrs
0018 204 CMPB FAB$B_RFM(R8),#FAB$C_FIX; fixed rfm?
0018 205 BNEQ 20$ ; branch if not
0018 206 RMSERR MRS ; anticipate problem
0018 207 MOVW FAB$B_MRS(R8),IFB$B_LRL(R9); set lrl
0018 208 BEQL MODXIT ; branch if zero (error)
0018 209 RMSSUC
0018 210 BRB MODXIT

06 1F A8 91 001D 195
    F2 1A 0021 196
    0023 197
03 1F A8 91 0023 198
    05 12 0027 199
5F A9 3F A8 90 0029 200
50 A9 1F A8 90 002E 201 10$:
60 A9 36 A8 B0 0033 202
    01 1F A8 91 0038 203
        OC 12 003C 204
        003E 205
52 A9 36 A8 B0 0043 206
    CB 13 0048 207
        004A 208 20$:
        C6 11 004D 209
            BRB MODXIT
```



```
004F 211
004F 212 :++
004F 213 : Escape type two - enable/disable echo of SYSS$INPUT to SYSS$OUTPUT
004F 214 :
004F 215 : Inputs:
004F 216 :
004F 217 :     ctx
004F 218 :
004F 219 : Outputs:
004F 220 :
004F 221 :     IFB$W_ECHO_ISI is changed to value specified.
004F 222 :
004F 223 : Notes:
004F 224 :
004F 225 :     1. FAB must describe SYSS$INPUT.
004F 226 :
004F 227 :     2. caller must not be user-mode.
004F 228 :
004F 229 :--
004F 230
004F 231 PPFECHO:
C2 69 2E E1 004F 232 BBC      #IFB$V_PPF_INPUT,(R9),MODXIT      ; not SYSS$INPUT
03      57 91 0053 233 CMPB      R7,#PS[$C_USER          ; user-mode?
      BD 13 0056 234 BEQL      MODXIT                ; that's a no-no
      1A A8 B0 0058 235 MOVW      FAB$L_CTX+2(R8),-
      2A A9      005B 236 IFB$W_ECHO_ISI(R9)          ; save stream's ISI
      B3 11 005D 237 RMSSUC
      0060 238 BRB      MODXIT
      0062 239 .END
```

```
$$PSECT_EP      = 00000000
$$RMSTEST       = 0000001A
$$RMS_PBUGCHK   = 00000010
$$RMS_TBUGCHK   = 00000008
$$RMS_UMODE     = 00000004
FABS$B_FSZ      = 0000003F
FABS$B_RFM      = 0000001F
FABS$C_FIX      = 00000001
FABS$C_MAXRFM   = 00000006
FABS$C_VFC      = 00000003
FABS$L_CTX      = 00000018
FABS$L_FOP      = 00000004
FABS$V_ESC      = 0000001B
FABS$W_MRS      = 00000036
FOP             = 00000020
IFBS$B_FSZ      = 0000005F
IFBS$B_RFMORG   = 00000050
IFBS$V_PPF_INPUT = 0000002E
IFBS$W_ECHO_ISI = 0000002A
IFBS$W_LRL      = 00000052
IFBS$W_MRS      = 00000060
MODXIT          = 00000015 R    01
PPFECHO         = 0000004F R    01
PSL$C_USER      = 00000003
RMSEX$RMS       = ***** X   01
RMS$FSET        = ***** X   01
RMESC_SETRFM    = 00000001
RMS$MODIFY      = FFFFFFFE RG   01
RMS$ENV         = 00018724
RMS$MRS         = 000185D4
RMS$RFM         = 00018664
SETRFM          = 00000018 R    01
```

```
+-----+
! Psect synopsis !
+-----+
```

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
RMS\$RMS	00000062 (98.)	01 (1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
\$ABS\$	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

```
+-----+
! Performance indicators !
+-----+
```

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.11	00:00:00.64
Command processing	137	00:00:00.74	00:00:06.80
Pass 1	237	00:00:05.93	00:00:15.84
Symbol table sort	0	00:00:00.66	00:00:00.97
Pass 2	55	00:00:01.21	00:00:03.14
Symbol table output	5	00:00:00.05	00:00:00.28
Psect synopsis output	1	00:00:00.02	00:00:00.11
Cross-reference output	0	00:00:00.00	00:00:00.00

Assembler run totals 473 00:00:08.72 00:00:27.79

The working set limit was 1350 pages.

32349 bytes (64 pages) of virtual memory were used to buffer the intermediate code.

There were 30 pages of symbol table space allocated to hold 620 non-local and 4 local symbols.

239 source lines were read in Pass 1, producing 13 object records in Pass 2.

19 pages of virtual memory were used to define 18 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

\$255\$DUA28:[RMS.OBJ]RMS.MLB;1
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1
\$255\$DUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)-----
9
1
4
14

739 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RMSOMODFY/OBJ=OBJ\$:RMSOMODFY MSRC\$:RMSOMODFY/UPDATE=(ENH\$:RMSOMODFY)+EXECML\$/LIB+LIB\$:RMS/LIB

0330 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

